



ELEKTRİK MÜHENDİSLERİ ODASI İSTANBUL ŞUBESİ
2004–2005 ÖĞRETİM YILI PROJE YARIŞMASI

UZAKTAN KOPYALA YAPIŞTIR

HAZIRLAYANLAR

Cem Başar ÇAYIROĞLU

Elif KAYA

Mehmet PEKMEZCİ

Proje Yöneticisi: Öğr. Gör. H.Turgut UYAR

Haziran 2007

1. GİRİŞ

Bilgisayar kullanım alanlarının ve grup çalışma gereksinimlerinin her geçen gün çok büyük bir hızla artması, beraberinde aynı anda birden fazla bilgisayarda aynı işlerin yapılmasını ve koordineli olarak ilerlemeyi zorunlu hale getirmiştir. Eşzamanlı olarak birden çok bilgisayarda çalışma esnasında ortaya çıkabilecek en önemli problemlerden birisi de bilgisayarlar arası küçük boyutlu metinlerin aktarımıdır. Bu işlemi gerçekleştirmek için ya ilgili metin bir metin dosyasına kopyalanarak dosya aktarımı gerçekleştirilmekte, ya da mail, Messenger gibi araçların kullanımı zorunlu kılınmaktadır. Özellikle aynı ağa bağlı olmayan ya da ağ bağlantısı bulunmayan bilgisayarlar arasında küçük metin aktarımları için bile büyük zaman kaybı yaratacak yöntemler şu an için tek çözüm yolu olarak görünmektedir.

Günümüzde bu probleme özel yaratılmış herhangi bir profesyonel çözüm bulunmadığı gibi bu güne kadar bu konu üzerinde yapılmış amatör çalışmaların dışında literatüre geçecek seviyede ciddi bir çalışma bulunmamaktadır.

Bu çalışmada bu soruna çözüm olması açısından iki bilgisayar arasında kopyala – yapıştır işlemini hızlı ve güvenli bir şekilde gerçekleştirmek üzere bir donanım ve ilgili yazılım tasarlanmış ve geliştirilmiştir. Proje geliştirilirken, hız, güvenlik, kullanım kolaylığı ve maliyet kriterleri göz önüne alınmıştır.

Projenin başarımlarını kriterlerini sağlayabilmesi adına bilgisayarlar arası kopyala – yapıştır işlemi aynı ağa bağlı bilgisayarlar arasında ağ üzerinden gerçekleştirilebildiği gibi, ağa bağlı olmayan bilgisayarlar arasında, bilgisayara USB arabirimden bağlanabilen bir donanım vasıtasıyla da gerçekleştirilebilecektir.

Ayrıca, farklı işletim sistemlerinde aynı şekilde çalışarak, maksimumda verim sağlamayı amaçlayarak, aynı amaca hizmet eden, aynı şekilde çalışan, birbirleriyle haberleşebilen, Windows işletim sistemi için C# dilinde .net platformunda geliştirilmiş, Linux işletim sistemleri için de Java dilinde Netbeans 5.5 platformunda geliştirilmiş programlarla proje gerçekleştirilmiştir.

Projede tamamen farklı iki farklı veri iletim şekliyle gerçekleştirilen haberleşme kullanıcı tarafından tamamen aynı şekilde kullanılabilir. Bu sayede bir kullanıcının rahatlıkla programı farklı ortamlarda farklı şekillerde kullanabilmesi sağlanacaktır. Ayrıca projenin güvenlik kriterleri doğrultusunda, kullanıcı kimlerle veri iletişimi yapacağı bilgisini kendisi belirlemek durumundadır. Kişi sadece izin verdiği kişilerle kopyaladığı verileri paylaşabilir, sadece kendisinin istediği ve kendisine izin veren kişilerin kopyaladığı verileri alabilir.

İki farklı bilgisayar arasında kopyala – yapıştır işlemini gerçekleştirmek üzere 1 numaralı kullanıcı herhangi bir dosyayı ya da herhangi bir metni kopyalar. 2 numaralı kullanıcı klavyedeki ilgili tuş kombinasyonundan kopyala komutunu verir. Bu komutla birlikte veri transferi gerçekleştirilir. Son olarak 2 numaralı kullanıcı normal şekilde veriyi yapıştır komutunu vermesiyle işlem tamamlanır.

Kablosuz bağlantı sistemleri, cihazların özgürlüğünü arttırdığı için gittikçe önemini arttırmakta ve yaygınlaşmaktadır. Söz konusu sistemlerin bu denli yaygınlaşması çok sayıda şirketi bu sektöre çekmiştir. Bir çok şirketin bu alanda çalışma yapmasından dolayı IEEE, IEEE 802.15.4 (Zigbee) standardını ortaya çıkarmıştır. Projede de kablosuz haberleşme için zigbee standartlarına uygun RF modemler kullanılmıştır.

Kısaca bu projenin amacı iki farklı bilgisayar arasında hem ağ bağlantısı üzerinden hem de ağ bağlantısı olmadan USB arabirimlerine bağlanmış olan birer donanım üzerinden bilgisayarların kopyala – yapıştır işlemlerini birbirleri arasında yapabilmeleridir. Böylelikle iki kullanıcıdan birisinin kopyaladığı bir metni bir diğeri yapıştırabilecektir.

Bu amaca yönelik izlenen yol ise öncelikli olarak projenin kapsamının belirlenmesi amacıyla projenin tanımlanması ihtiyacı belirlenmiştir. Bu sebeple hiçbir işlem başlamadan önce proje tanımının yapılması, ardından projenin analizine geçilmesi planlanmıştır. Projenin analizi kısmında projenin hangi ana modüllerden oluşacağı, farklı işletim sistemleri üzerinde en yüksek performansı vermesi için hangi programlama dillerinin kullanılacağı ve donanımsal kısmın projenin hangi aşamasında ve ne şekilde yer alacağı belirlenmek istenmiştir. Projenin analizi tamamlandıktan sonra projeye temel olacak kaynak tarama işlemlerine geçilmesi, projeye temel olabilecek dokümanlar belirlenmesi planlanmıştır. Dokümanların belirlenmesi aşamasından sonra RF haberleşmeyi gerçekleştirecek olan modülün özelliklerinin ne olması gerektiği ve piyasada bu özelliklere sahip hangi donanımların bulunduğu araştırılması ve uygun modülün seçilmesi planlanmıştır. Bu işlemler gerçekleştirildikten sonra bilgisayarlar arası haberleşmenin gerçekleşeceği protokolün belirlenmesi aşamasına geçilmesi planlanmıştır. Bu aşamadan sonra ancak donanımsal ve yazılımsal tasarımın yapılabileceği belirlenmiştir. Son olarak test, bakım ve dökümantasyon aşamalarına geçilmiştir.

KURAMSAL BİLGİLER

Kablosuz iletişim bir yerden bir yere bilgi transferinin arada elektriksel iletkenler ya da kablolar olmadan gerçekleştirilmesidir. Genellikle mobil bilgi teknolojilerinde kullanılır. Cep telefonları, PDA lar, kablosuz ağ, GPS birimleri, garaj kapıları, kablosuz fare ve klavye, uydu televizyonları, telsiz telefonlar kablosuz iletişime örnek verilebilir.

Uzun mesafelerde kurulan kablosuz çalışmaların kablo kullanarak yerine getirilmesi imkansızdır ya da pratik değildir. Kablosuz haberleşmede enerjinin bazı formları kullanarak bilgi taransferi gerçekleştirilir. Bunlar radyo frekansı(RF), kızılötesi ışık, lazer ışık, görünür ışık, akustik enerji vb...' dir. Kablosuz haberleşme:

- Radyo frekansıyla
- Mikrodalga haberleşmesiyle(kısa mesafelerde)
- Kızılötesi (IR) kısa mesafe iletişim(uzaktan kumanda gibi) ile gerçekleşir.

Uygulamalar bir noktadan bir noktaya ya da bir noktadan daha çok noktaya(radyo tv yayınları, cep telefonları, kablosuz ağ) iletişimi içerir.

Kablosuz İletişim Standartları

IEEE, kablolu ağlarda olduğu gibi, kablosuz ağlarda da standartlar konusunda bir numaralı kuruluştur. Bu standartlar FCC (Federal Communications Commission) tarafından yapılan düzenlemelerle aynı çerçeve yapısında çalışacak şekilde oluşturulmaktadır.

Radyo iletişiminde fiziksel bazı kurallar vardır. Gerekli standartlar oluşturulurken bu özellikler göz önüne alınmaktadır. Eğer yüksek hızda bir iletim gerçekleştirmek istenirse verinin tekrarlanmadan gönderebileceği alan düşük olmaktadır. İletim alanını arttırmak için güç tüketimini arttırabilir, ama bu durumda da cihazın pil ömrü azalmaktadır. Bu nedenle taşınabilir cihazlara kablosuz bağlantı özelliğini katacak RF biriminin seçiminde, hız ve kapsama alanı özellikleri kadar güç tüketim özelliği de önem kazanmaktadır. İletim hızını arttırmanın diğer bir yolu da daha yüksek frekanslarda iletimi gerçekleştirmektir. Bu durumda da tekrarlamasız iletim alanımız kısalmaktadır. Özetleyecek olursak:

Yüksek veri oranları =Kısa iletim alanı

Yüksek güç tüketimi =Artan iletim alanı, kısalan pil ömrü

Yüksek frekansta iletişim =Daha yüksek veri hızları, daha kısa iletim alanı.

Yukarıda bahsedilen iletim kuralları ve insanların ihtiyaçları baz alınarak IEEE tarafından aşağıda bahsedilen standartlar tanımlanmıştır. Ve bu standartlar üzerinde çalışmalar hala devam etmektedir.

Kablosuz LAN standartları genel olarak 4 tanedir:

- 802.11
- 802.11b
- 802.11a
- 802.11g

	802.11b	802.11a	802.11g
Frequency Band	2.4 GHz	5 GHz	2.4 GHz
Availability	Worldwide	US/AP	Worldwide
Maximum Data Rate	11 Mbps	54 Mbps	54 Mbps

WLAN standartları

802.11

IEEE tarafından 1997 yılında tanımlanan ilk kablosuz iletişim standardıdır. 802.11 diğer tüm varyasyonlarının tabanı olan basit bir kablosuz teknoloji standardıdır. 1.2 Mbps hızına sahiptir. Çalışma mekanizması ise DSSS(Direct Sequence Spread Spectrum) ve FHSS(Frequency Hopped Spread Spectrum)' dir.

Alan	Frekans (GHz)	
USA	2.4000	2.4835Ghz
Avrupa	2.4000	2.4835Ghz
japonya	2.471	2.497Ghz
Fransa	2.4465	2.4835Ghz
İspanya	2.445	2.475Ghz

WLAN Çalışma Frekansları

802.11b

802.11 standardı üzerinde ilk büyük değişiklik IEEE tarafından 1999 yılında yapılmıştır. Wi-fi teknolojisinde bu değişiklik kullanım kolaylığı, kurulum, esneklik ve göreceli fiyat bakımından bir sıçrama olarak duyuruldu.

Çalışma Frekansı	2.4 GHz
İletim Hızı(teorik)	1, 2, 5.5, 11 Mbps
İletim Hızı(ölçülen)	4 Mbps(ortalama)
Mekanizma	DSSS
Kanal Sayısı	11(örtüşme olmadan)
Maksimum Uzaklık	175 ft(ortalama)

802.11b Standartları Özellikleri

Avantaj Dezavantaj Analizi: 11b'nin göze çarpan ilk dezavantajı 2.4 Ghz'de çalışmasıdır çünkü 2.4Ghz bandı çok kalabalık olmaya başlamıştır. Bu bantta çalışan kablosuz veri iletimleri birçok değişik sinyalle(mikrodalga fırın, kablosuz telefonlar, Bluetooth cihazları..v.b. gibi gene bu bantda çalışan cihazların sinyalleri ile girişim yapabilmektedirler. Gürültü oranı artabilir bu da işaret gürültü oranını düşürerek haberleşme kalitesini azaltır.

802.11a

802.11 standardı üzerinde ikinci büyük değişiklik IEEE tarafından 2001 sonlarına doğru yapılmıştır. 802.11a dikkat çekici bir hız artışı (11b'nin nerdeyse 5 katı) ile ortaya çıkmıştır.

Çalışma Frekansı	5.8 GHz
İletim Hızı(teorik)	54 Mbps' a kadar
İletim Hızı(ölçülen)	20-36 (yaklaşık ortalama)
Mekanizma	OFDM(Ortogonal frekans bölme çoğullayıcı)
Kanal Sayısı	12(örtüşme olmadan)
Maksimum Uzaklık	80 ft(ortalama)

802.11a Standartları Özellikleri

Avantaj Dezavantaj Analizi: Bu sistemi kullanmak biraz maliyetli olabilmekte çünkü aynı alanı kaplamak için 11a'da daha fazla erişim noktasına ihtiyaç duyulmaktadır. Ama 11b'ye göre kurulum, yapılandırma ve erişim noktası yerleştirme açısından (12 tane ayrı üst üste binmeyen kanaldan oluştuğu için) daha basittir. Ayrıca 11b'ye geriye doğru bir destek sağlamamaktadır.

802.11g

802.11g, mevcut sorgulama halindeki IEEE standardıdır. 11g'nin, 11a ve 11b'nin bazı özelliklerini birleştiriyor olması onu eşsiz kılmaktadır. 11g, 11b ile aynı çalışma frekansını kullanmakta ve CCK kullanarak 11b ile 5.5Mbps ve 11Mbps hızlarında uyumlu olarak çalışabilmektedir. Bunun yanında 11g, 11a'nın OFDM teknolojisini alarak 54Mbps hızında da çalışmaktadır.

Çalışma Frekansı	2.4 GHz
İletim Hızı(teorik)	54 Mbps
İletim Hızı(ölçülen)	20-30 (ortalama)
Mekanizma	OFDM, CCK(Complimentary Code Keying)
Kanal Sayısı	3 (1, 6, 16)
Maksimum Uzaklık	175 ft(ortalama)

802.11g Standartları Özellikleri

Avantaj Dezavantaj Analizi: 11g maksimumum uzaklığına yaklaştığı zaman 11a gibi hala yüksek iletim hızı sağlayamamaktadır. 2.4 Ghz'de çalıştığı için çok fazla girişime maruz kalmasına rağmen firma veya kullanıcılar açısından çok çeşitli yararlar sağlamaktadır. 11b ile uyumlu olarak çalışması çok önemli bir özelliğidir. Eskiden kurulmuş sistemlerin kolayca yenilenmesi ve güncellemesi yapılabilmektedir. 11g, 11a'ya göre daha ucuz bir alternatiftir. Sonuç olarak 11g, 11b ve 11a ile karşılaştırıldığında kendisine fiyat performans(yüksek hız ve göreceli düşük fiyat) açısından ayrıcalıklı bir yer bulmuştur.

Ek Standartlar:

- 802.11c : Bu standart erişim noktaları arasında köprü kurmakla sorumludur.
- 802.11d: 802.11 standartlarının yaygınlaşması ve kabul görmesi ile ilgilenmektedir.
- 802.11f: Değişik üreticiler tarafından üretilen erişim noktalarının arasındaki uyumluluğu inceleyip çözümlemektedir.
- 802.11i: Ana odaklanma noktası kablosuz güvenlidir.
- 802.11e: Kablosuz iletim alanında görüntü ve ses iletimi ile ilgilenmektedir.
- 802.11n: 802.11' in hızını "fast ethernet"(minimum 100Mbps) hızına çıkarmayı amaçlamaktadır.

Kablosuz haberleşme ile ilgili bütün bu çabalar çok hızlı biçimde sonuçlarını vermektedir. Bugün birçok üniversite kampüsünde, ülkelerin ünlü sokaklarında, işyerlerinde kablosuz

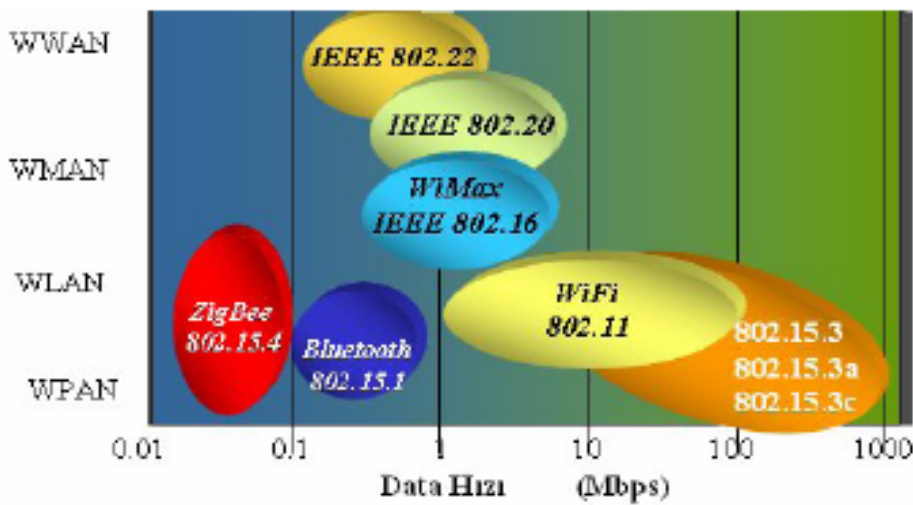
internet erişimi bulunmaktadır. Hatta ADSL teknolojisi sayesinde birçok evde de kablosuz ADSL modemler kullanılarak internet erişimi sağlanmaktadır. Su altında kurulan, her bir hücresi kibrit kutusundan küçük boyutta olan hareketli kablosuz ağlar(ad-hoc networks) depremle ilgili sismik veriler toplayıp ana merkeze iletebilmektedir.

Kablosuz haberleşme sistemi, kablolu ağların oluşturulamadığı ortamlarda iletişimi sağlamakta bu sayede çalışma verimini arttırmaktadır. Ayrıca ölçeklenebilirliği sayesinde de iletişimci sayısının genişlemesini kolaylıkla destekleyebilmektedir. Yukarıda bahsedilen avantajlarından dolayı kablosuz haberleşme alanında bilimsel çalışmalar hızla devam etmekte ve meydana gelen gelişmeler göz önüne alındığında bu alanda gidilebilecek son noktanın sadece hayal gücü ile sınırlı olduğu görülmektedir.

ZIGBEE

Zigbee Alliance güvenilir, düşük maliyetli, düşük güç kullanılan, telsiz ağ iletişimi için izleme ve kontrol ürünleri üzerinde çalışan şirketlerden oluşan bir kurumdur. ZigBee Alliance üyeleri ortak çalışarak zigbee standartları ve özelliğini belirliyor ve ilerlemeyi sağlamaktadır. Zigbee Alliance'a dahil katılımcılardan bazıları; Ember Corporation, Honeywell, Invensys, Mitsubishi Electric, Motorola, Philips ve Samsung. Bundan başka 100 'ü aşkın katılımcıları bulunmaktadır.

Zigbee standardının iki önemli avantajı vardır. Bunlardan birincisi bu standardın düşük güç tüketimi ve düşük maliyet ilkesi üzerine tasarlanmış olmasıdır. Güç açısından verimli olmasının ilk sebebi düşük data hızı ile çalışmasıdır. Şekil 2.1'e bakılarak zigbeenin data hızının diğer kablosuz bağlantı türlerinin data hızlarıyla karşılaştırılması yapılabilir. İkinci sebep ise cihazların aktif olmadığı durumlarda uyku moduna geçirilip yaklaşık 3 mikroamper gibi düşük bir akım ile çalıştırılabilir olmasıdır. Zigbee uygulamalarında hafıza gereksiniminin az olması nedeniyle maliyet düşük olmaktadır. Zigbee standardının ikinci avantajı ise boyutlarının küçük olmasıdır.



Data Hızı – Ağ Genişliği Grafiğinde 802.15.4 Standardının Yeri

Kısa mesafe kablosuz ağ standardı olan Zigbee 2,4 Ghz frekans bandını kullanmaktadır. 868-915 Mhz mertebesinde 11 kanalı ile 40 Kb/s; 2,4 Ghz frekansta ise 16

kanalı ile 250 kbps veri iletim hızına ulaşabilmektedir. Uyku Modu özelliği sayesinde aktif olmadığı zamanlarda uykuya geçmekte ve yüksek enerji tasarrufu sağlamaktadır. Örneğin nadiren ortaya çıkan bir durumu algılayan sensörün durum değişikliğini bir kontrolöre iletecek olan Zigbee aygıtı kalem pille 2 yıl kadar çalışabilmektedir. Önemli özelliklerinden biri de herbir zigbee aygıtının hem alıcı hem verici olarak çalışabilmesidir.

Ayrıca erişim mesafesi 10-70 mt civarı olan aygıt, ortamdaki diğer zigbee aygıtlarını access point olarak kullanarak uzak mesafelere veriyi taşıyabilmektedir. ZigBee standardı; uzaktan görüntüleme, kontrol ve algılama ağları uygulamalarının gereksinimlerini belirleyen tek teknoloji esaslı standarttır. Anlatılan özellikler tablo 2.1 de verilmektedir.

IEEE 802.15.4 Zigbee Standartları

Ultra düşük maliyetli ve ultra düşük güç tüketimli iletişim alt yapısı olarak IEEE 802.15.4, uzun pil ömrüne imkan tanıyan fakat dolayısıyla düşük veri hızları sunan bir standarttır [2]. Zigbee standartları ile ilgili detaylı bilgiler tablo 2.1’de verilmiştir. Şekil 2.2’den de görüleceği gibi zigbee standardı IEEE 802.15.4 global standartına, uygulama, güvenlik ve ağ katmanları eklenerek oluşturulmuş bir standarttır.

Zigbee standartları

Özellik	Aralık
Data Hızı	868 MHz-20kb/s, 915 MHz-40kb/s, 2.4 GHz-250kb/s
Ünite Sayısı/Ağ	255
Mesafe	10 - 75 m
Karmaşıklık	Mevcut standartlardan daha az karmaşık
Batarya Ömrü	6 Ay – 2 Yıl
Kanal Sayısı	868, 915 MHz – 11 Kanal, 2.4 GHz – 16 Kanal
Adres	8 bit veya 64 bit
Sıcaklık	-40 ... 85 °C

RF Modem Özellikleri

- Zigbee standartlarına uygun olan bir alıcı-verici QPSK modülasyonunu kullanarak 5 MHz kanal bant genişliğinde 250 kb/s hızına kadar çıkabilmektedir.
- 2.4 GHz ISM (Endüstriyel, Bilimsel ve Tıbbi) bandı üzerindeki 16 seçilebilir kanallardan bir tanesi üzerinden çalışır.
- -1 dBm den 0 dBm’e kadar nominal çıkış gücü ve -27 dBm den +3 dBm’e kadar programlanabilir tipik çıkış gücü bulunmaktadır.
- Zigbee standartı olan -85 dBm den daha iyi bir duyarlılık göstermektedir. (< -92 dBm %1’lik PER paket hata oranı)
- Alıcı vericiyi saturasyona uğratan maksimum giriş seviyesi 10 dBm’dir.

Başarılı Veri Transferi Olasılığını Arttırmak İçin Kullanılan Yöntemler

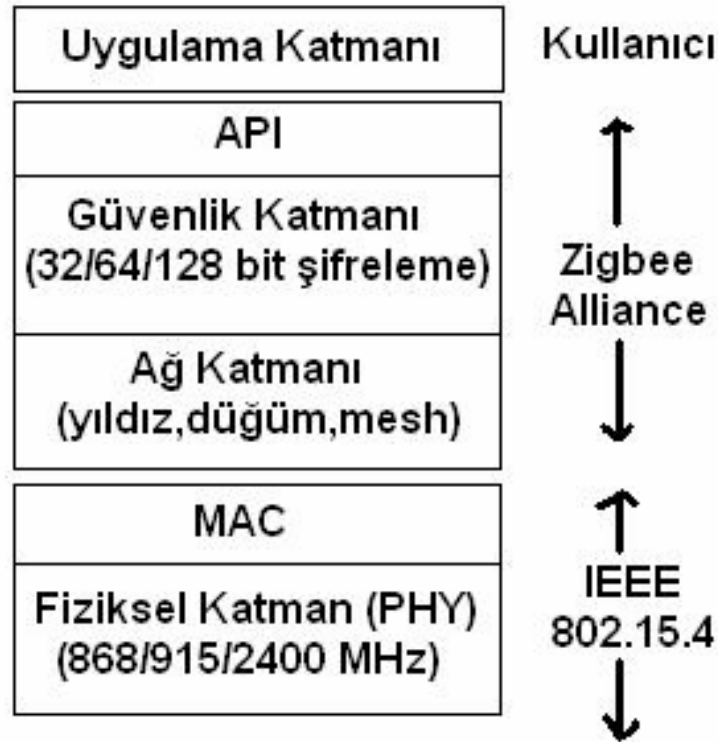
CSMA-CA mekanizması : Bu tip mekanizmada cihaz veri transferine başlamadan önce rasgele belirlenen bir süre bekler. Bu süre içerisinde veri transferi yapılacak olan kanal boş ise veri kanala bırakılır. Dolu olması durumunda cihaz farklı bir rasgele süre daha bekler ve kanalın boşalmasından sonra işlemini gerçekleştirir. Bu aynı kanalda aynı anda birden fazla veri olmasını engelleyen bir yapıdır.

Veri Tasdifi: Bu mekanizmada ise gönderici taraf alıcı taraftan veri geldi onayı beklemektedir. Alıcı taraftan bu şekilde onay alınmadığı takdirde o gönderimin başarısız olduğu kabul edilerek yeni bir gönderim denenir. Bu şekilde veri güvenliği sağlanmaya çalışılmaktadır. Gönderim denemeleri belirli bir sayı sonra başarısız olursa, gönderici taraf bağlantıyı sonlandırır veya tekrar dener.

Veri Doğrulaması: Bu mekanizmada ise her bir veri katarındaki hataların tespiti için ITU standardındaki 16 bitlik CRC (Cyclic Redundancy Check) algoritması kullanılmaktadır.

Protokol Yığın Yapısı

Zigbee'nin ağ katmanlarında ilk iki sırayı PHY(Fiziksel Katman) ve MAC (orta düzey erişim kontrolü) alır. Yapılacak uygulama çok özel bir uygulama değil ise bu katmanlarda bir değişiklik yapmaya gerek yoktur.



Zigbee protokol standardı içinde katmanlar ve özellik belirleyicileri

Aşağıdaki tabloda zigbeenin diğer kablosuz teknolojilerle anahtarlama yapıları açısından karşılaştırılması verilmiştir.

Kablosuz teknolojilerin anahtarlama özellikleri bakımından karşılaştırılması

	Zigbee	Bluetooth	UWB	Wi-Fi
Standart	IEEE 802.15.4	IEEE 802.15.1	IEEE 802.15.3a	IEEE 802.11 a,b,g
Endüstri Organizasyonu	Zigbee Alliance	Bluetooth SIG	UWB Forum & WiMedai Alliance	Wi-Fi Alliance
Yapı	Mesh, Yıldız, Düğüm	Yıldız	Yıldız	Yıldız
RF Frekansı	868/915 MHz 2.4 GHz	2.4 GHz	3.1-10.6 GHz	2.4 GHz 5.8 Ghz
Veri Hızı	250 Kbps	723 Kbps	110 Mbps – 1.6 Gbps	11-105 Mbps
Menzil	10 - 70 m	10 m	4 - 20 m	10 – 100 m
Güç	Very Low	Low	Low	High
Pil Ömrü	Alkalin (ay - yıl)	Şarz edilebilir (gün - hafta)	Şarz edilebilir (saat - gün)	Şarz edilebilir (saat)
Düğüm Sayısı	65000	8	128	32



Üstünlük Yönleri

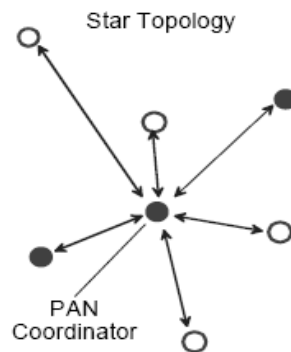


Zayıf Yönleri

İncelendiğinde zigbee 802.15.4 standardının gönderilecek veri yoğunluğu bakımından biraz zayıf kaldığı görülebilir. Ancak topoloji(yapı), range(menzil) ve güç bakımından üstünlük sağlamaktadır. Ayrıca yüksek node sayısı özelliği ile kablosuz ağ alanında bir çok uygulama alanında kullanılabileceği konusunda kendini kanıtlamıştır.

Zigbee Ağ Yapıları

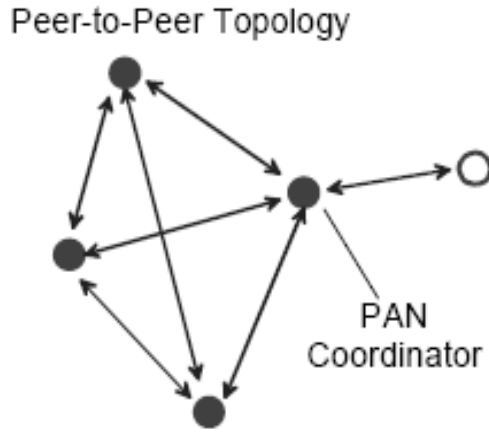
1.Yıldız Ağ Yapısı



Zigbee Yıldız Ağ Yapısı

Yıldız ağ yapısı haberleşmesi tek bir merkez birim ile uç birimler arasında gerçekleşmektedir. Bu merkez birim üzerindeki yazılımlar ile network üzerinde haberleşmeyi başlatma, yönlendirme veya durdurma görevlerini tek başına üstlenir. Sistemdeki tüm birimler 64 bit adresleme ile ayrıştırılırlar.

2. D ğ mden D ğ me Ađ Yapısı



Bu yapının yıldız ađ yapısından farkı, t m birimlerin birbirleri ile kapsama alanı dahilinde haberle ebilmeleridir. End striyel otomasyon ve izleme, kablosuz sens rler, akıllı tarım ve g venlik bu tip ađ yapılarının kullanıldığı belli ba lı alanlardır.

IEEE 802.15.4 PHY

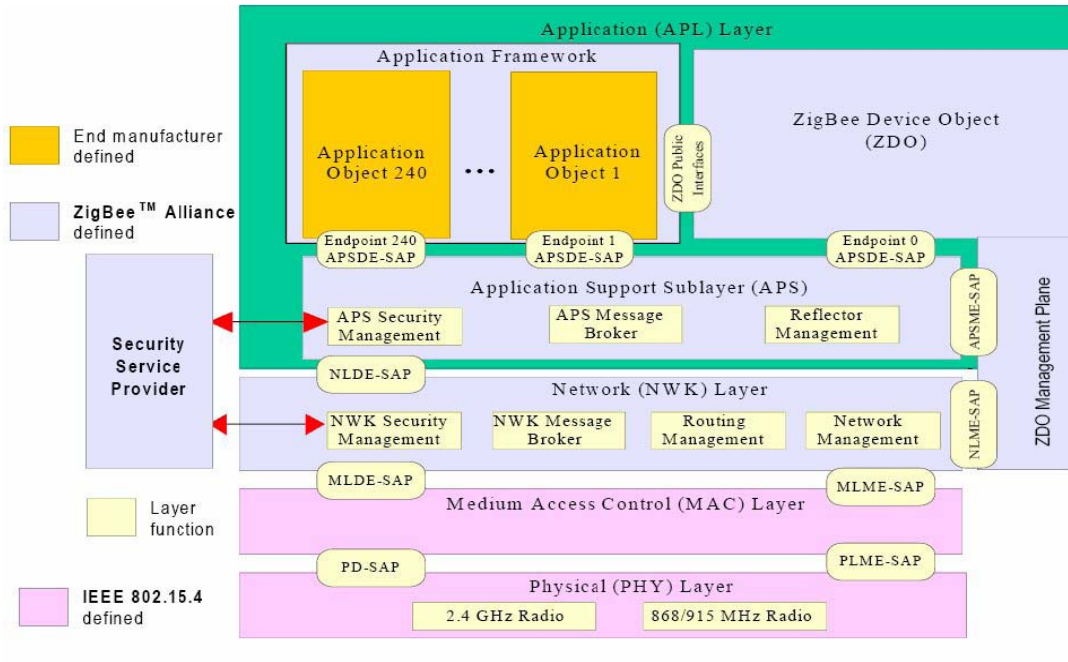
Radyo alıcı-vericileri i in aktivasyon ve deaktivasyon, Enerji Taraması, Bađlantı Kalite  l  m , Kanal Se me, Temiz Kanal Deđerlendirmesi ve paketlerin MAC ile ileti imi bu katmanda ger ekle ir.

IEEE 802.15.4 MAC

MAC, i aretci (kablosuz ađlarda istemcilerin dinlemeye ge mesi i in g nderilen uyarı i areti) y netimi, kanal eri imi, frame g ncellemesi ve PAN (ki isel alan ađı) koordinat r  ile ili kilendirme fonksiyonlarının ger ekle tiđi katmandır.

Zigbee Ađ Katmanı- NWK

Zigbee standardında bahsedilen  ođu avantaj bu katman  zerinde ger ekle tirilen uygulamalar ile anlam kazanır. Bu katman i in olu turulacak uygulama yazılımıyla hem ađın parametreleri olu turulur hem de rotalandırma algoritması tasarlanır. Zigbee NWK dizaynı g m l  ađ sistemleri i in d   k maliyet ve d   k g   t ketimi demektir. Burada bahsedilen d   k maliyet, cihaz i in k   k bellek alanı (ve fiziksel boyut), d   k g   ise uzun pil  mr  anlamında deđer kazanır.



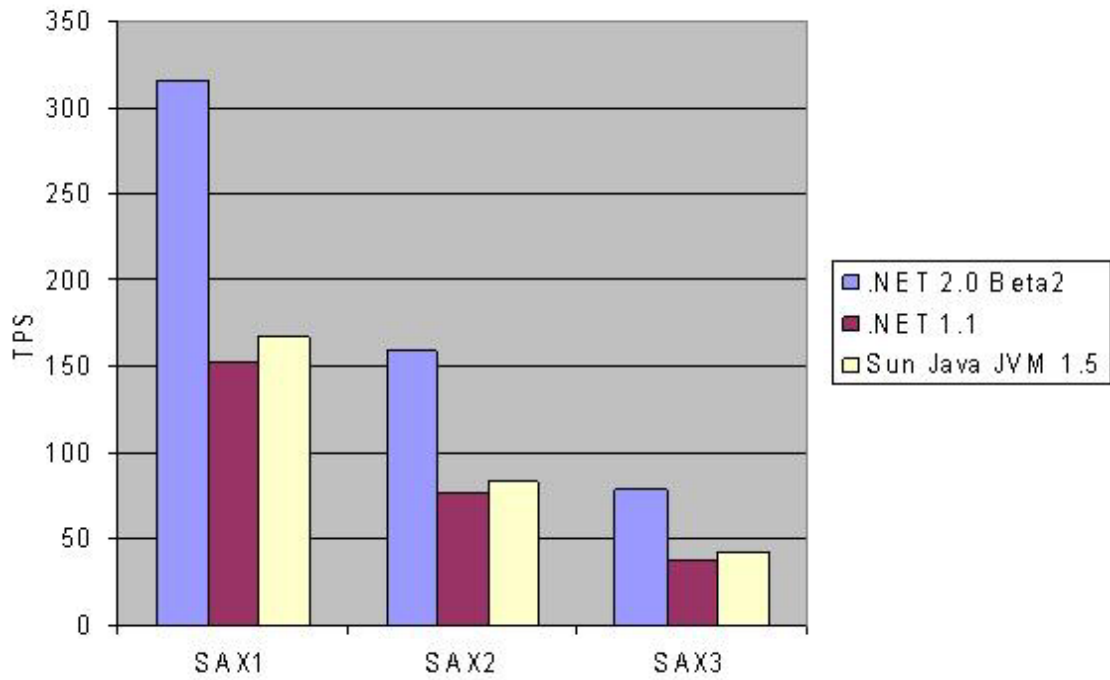
Zigbee Yığın Mimarisi ve Katmanlar Blok Diyagramı

ANALİZ VE MODELLEME

Projenin tasarımı için kullanılacak olan donanımsal ekipmanların ve programlama dillerinin belirlenmesi için farklı olasılıklar göz önüne alınarak analiz gerçekleştirilmiş ve tasarım aşamasına geçiş gerçekleştirilmiştir.

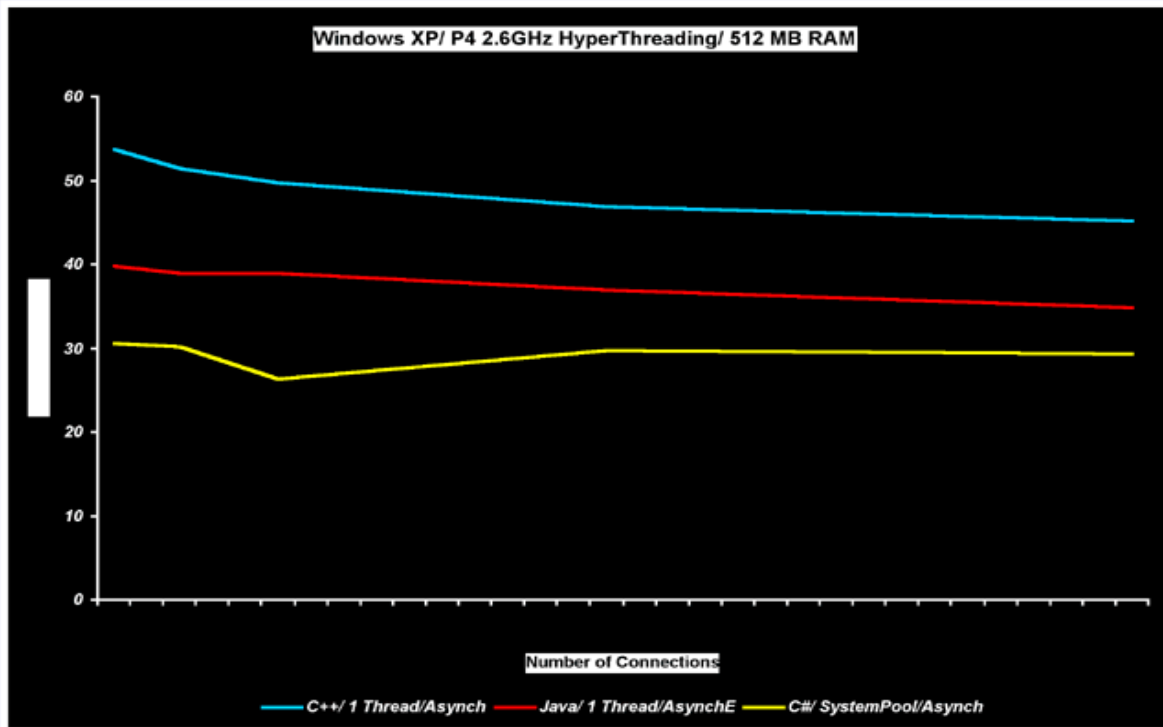
Yazılımsal Analiz

Projenin farklı işletim sistemleri üzerinde maksimum verimle çalışabilmesi açısından ilk önce hangi programlama dilinin kullanılmasının daha efektif olacağının kararı verilmiştir. Bu aşamada öncelikli olarak gelişmiş platformlara sahip, kullanımı daha kolay ve aynı zamanda efektif olan C# ve Java dilleri üzerinde kıyaslama yapılmıştır. Linux işletim sisteminde her ne kadar C# dilini kullanmak ayrı bir iş yükü getirecek olsa da Windows üzerinde yüksek performans gösterdiği yapılan analizler sonucu belirlenmiştir. Özellikle XML dosyalarının işlenmesi aşamasında MSDN web sayfasında yapılmış analize göre .NET 2.0 framework ile birlikte C# dili Java'dan daha iyi sonuçlar verebilmektedir.

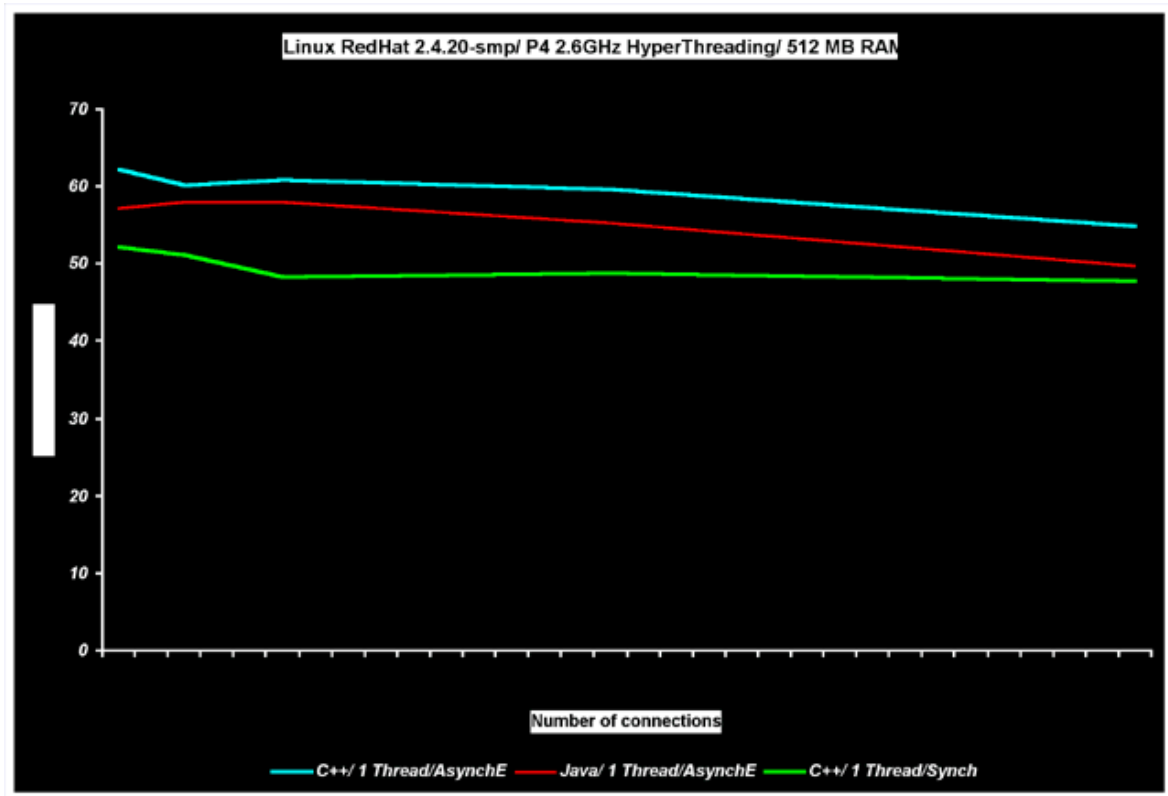


Java ve C# XML dosyası işleme performans analizi

Bunun yanı sıra Java programlama dilinin giriş çıkış işlemlerinde C#'a göre daha yüksek performans sağlamış olduğu yapılan araştırmalar sonucunda belirlenmiştir. Libman ve Gilbourn'un yapmış oldukları performans testlerine göre hem Linux hem de Windows işletim sistemlerinde Java programlama dilinin daha yüksek performans sağlamış olduğunu belirlemişlerdir. Aşağıdaki şekillerde test sonuçları görülebilir.



Java ve C# dilleri giriş çıkış işlemleri Windows XP performansları



Java ve C# dilleri giriş çıkış işlemleri Windows XP performansları

Bunların dışında analiz aşamasında göz önünde bulundurulacak diğer bir faktör de JVM (Java Virtual Machine) ve sürümü olmuştur. Programın sistem tray’de çalışması planlanmıştır. Bu işlem ancak Java 1.6 sürümü ile desteklenmektedir. Bu durum, Java dili kullanılarak yazılan programın çalıştırılabilmesi için JRE 1.6’nın kurulu olması anlamına gelmektedir.

Yapılmış olan araştırmalar ve bireysel tecrübelerimiz göz önüne alındığında Windows işletim sistemlerinde C# dilini Linux işletim sistemlerinde de Java dilini kullanmanın en uygun çözüm olacağı belirlenmiştir.

Donanımsal Analiz

Donanım kısmında göz önünde bulundurulacak en önemli analiz konuları RF iletişimi gerçekleştirilecek modülün belirlenmesi ile RF modül ile PC arasındaki iletişimin nasıl bir yazılımla gerçekleşeceğinin belirlenmesidir. Freescale firmasından temin edilen zigbee uyumlu alıcı verici ünitelerinin en verimli şekilde kullanılabilmesi için yine freescale firmasına ait 8bitlik HCS08 mikroişlemcisi kullanılmıştır. Bu mikroişlemcinin yazılımı c dili ile yazılmıştır ve firmanın göndermiş olduğu gerekli API’ler kullanılmıştır.

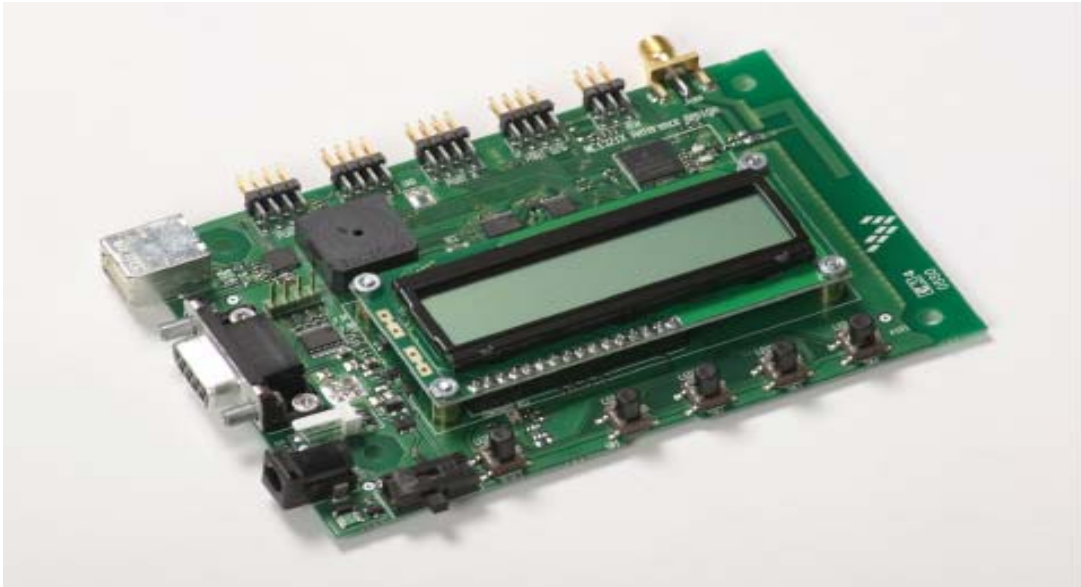
Radyo frekansı ile haberleşme işlemini gerçekleştirmek üzere kullanılması planlanan modülün piyasadaki eşdeğerleri arasından Freescale firmasının ürettiği alıcı-verici ve mikrokontrolör kısımlarını içeren 13213 SIP (system in package) çipi tercih edilmiş, bu çipi

içeren ve örnek uygulamaların çalıştırılabileceği 13213 NSK (network starter kit) geliştirme kiti temin edilmiştir. Kit NCB ve SRB olmak üzere iki tip cihaz içermektedir.

NCB

NCB, 13213, 2.4GHz alıcı-verici entegresi içeren 802.15.4/Zigbee geliştirme kartıdır. NCB, bilgisayar ile hem USB hem de RS232 UART bağlantı seçenekleri sunmaktadır. Üzerinde 16x2 LCD ekran ve buzzer bulunmaktadır. NCB aşağıdaki arayüzleri içermektedir:

- 2.0 USB port
- 4 adet buton (SW1, SW2, SW3 ve SW4)
- 1 adet reset butonu
- 4 adet LED (LED1, LED2, LED3 ve LED4)
- 1 adet açma-kapama anahtarı
- Baskı devre üzerinde F anten
- BDM modülü
- Besleme girişi
- Batarya



NCB geliştirme kartı

SRB

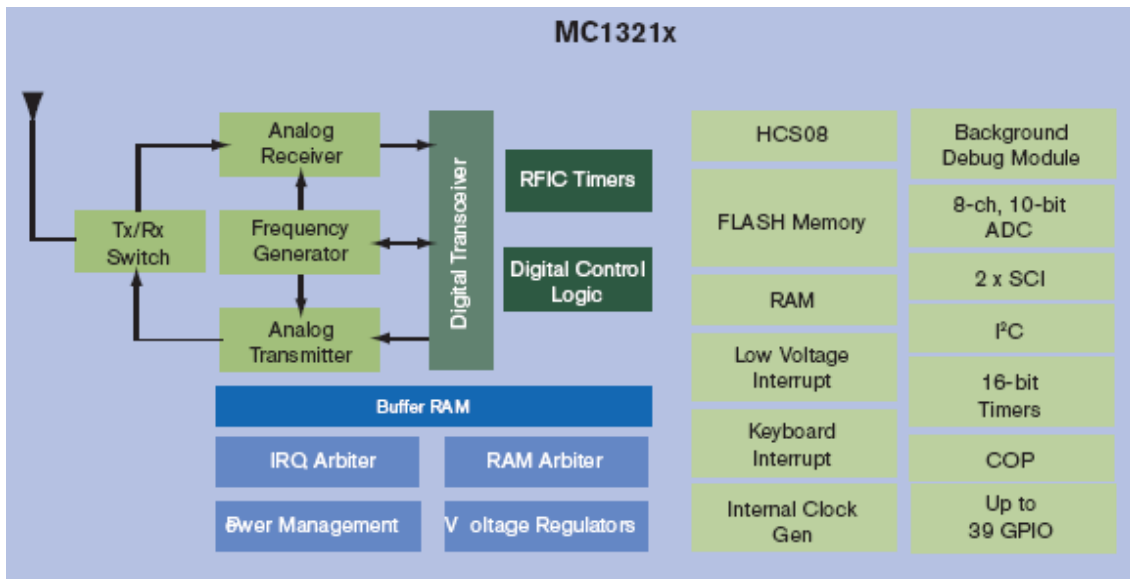
NCB, 13213, 2.4GHz alıcı-verici entegresi içeren 802.15.4/Zigbee geliştirme kartıdır. SRB bilgisayar ile sadece USB bağlantı seçeneği sunmaktadır. LCD ekran bulundurmamaktadır. SRB aşağıdaki arayüzleri içermektedir:

- 2.0 USB port
- 4 adet buton (SW1, SW2, SW3 ve SW4)
- 1 adet reset butonu

- 4 adet LED (LED1, LED2, LED3 ve LED4)
- 1 adet açma-kapama anahtarı
- Baskı devre üzerinde F anten
- BDM modülü
- Besleme girişi
- Batarya
- Mikrokontrolör ve alıcı-verici için 26 adet pin



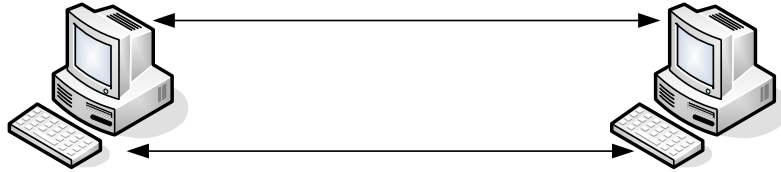
SRB geliştirme kartı



MC13213 alıcı-verici entegresinin blok şeması

Modelleme

Modelleme gerçekleştirilirken her bir PC'nin aynı zamanda hem birer server hem de birer istemci olarak davranacakları belirlenmiştir. Bu sayede her bir bilgisayar hem server rolü üstlenerek hattı dinlemekte hem de kullanıcıdan istek geldiğinde istemci rolünde karşı taraftan veri talebi gerçekleştirmektedir.



Bilgisayarlar ve üstlendikleri roller

Kullanıcılar hangi bilgisayarlarla veri paylaşımında bulunacaklarını kendileri belirleyeceklerdir. Böylelikle bir XML dokümanında saklanan, bilgi paylaşılan kullanıcılar listeleri bilgisayarlar arası haberleşmeye getirilmiş olan tek kısıttır.

Ayrıca kullanıcılar ağ üzerinden mi donanım üzerinden mi haberleşeceklerini kendileri belirlemek durumundadırlar.

Client

Tasarım, Gerçekleme ve Test

Projede katmanlı yapı kullanılarak soyutlamalara ve standartlaşmaya gidilmiştir. Sistemde sadece ardarda olan katmanlar arasında veri aktarımı gerçekleştirilmektedir. Aynı katmanlar farklı işletim sistemleri için farklı dillerde yazılmış olmasına rağmen, tasarım aşamasında her katmandaki her program parçacığının ne amaca hizmet edeceği ve hangi diğer parçacıkları ile veri alışverişi içerisinde olacağı belirlenmiş olduğu için, farklı platformlarda farklı dillerde çalışan uygulamalarda belirli bir standartlaşma sağlanmıştır. Buna göre en temel halde projenin katmanları aşağıdaki şekilde gibidir:

Server

Kullanıcı Arayüz Katmanı	FROM1 KULLANICIDAN GELEN KLİPBORD İSTEKLERİNİ SERVER VE CLIENT NESNELERİNİ KULLANARAK GERÇEKLER		FROM2 KULLANICININ KLİPBORD ALIM VE GÖNDERİM LİSTESİNİ DÜZENLEYECEĞİ BİR ARAYÜZ OLUŞTURUR DOĞRUDAN 3. KATMAN İLETİŞİM KURAR
İletişim Katmanı	SERVER KLİPBORD İÇERİK İSTEĞİNDE BULUNAN BİLGİSAYARLARA CEVAP VERİR IP LİSTESİNE BAKARAK İSTEĞE CEVAP VERİP VERMEYECEĞİNE KARAR VERİR 3. KATMANDAKİ NESNE İLE İLETİŞİMDİR	CLIENT KLİPBORD İÇERİK İSTEĞİNDE BULUNUR.İSTEKTE BULUNACAĞI IP YI 3. KATMANDAKİ NESNE İLE ÖĞRENİR	CONFIGURE TÜM KONFIGÜRASYON BİLGİLERİNİN 3. KATMANA AKTARILMASINDAN SORUMLUDUR.
Kayıt Katmanı	IPSTORER IP KAYITLARI İLE İLGİLİ BÜTÜN İŞLEMLERDEN SORUMLUDUR.		
XML VERİTABANI	IP.XML DOKÜMANI		

Katmanlı Yapı

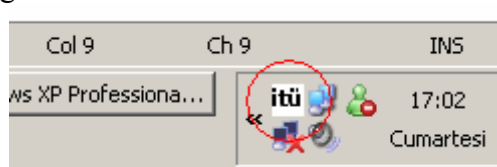
Kullanıcı Arayüz Katmanı

Kullanıcı arayüz katmanı kullanıcıyla program arasında veri akışının gerçekleştiği tek kısımdır. Bu katmanda kullanıcı:

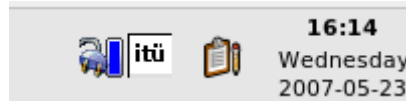
1. Program ayarlarını görüntüleyebilir / değiştirebilir:
 - a. Hangi yöntemle iletişim kurulacağını görebilir / belirleyebilir. (Ağ üzerinden / Donanım üzerinden)
 - b. Kimlerden veri alacağını görebilir / değiştirebilir.
 - c. Kimlere veri aktaracağını görebilir / değiştirebilir.
 - d. Kendi ismini görebilir / değiştirebilir. (Sadece donanım üzerinden iletişim gerçekleştirildiğinde)
2. Uzaktan kopyalama işlemini başlatabilir.

Kullanıcı Arayüz Katmanı Kullanım Şekli

Kullanıcı uygulamayı çalıştırdığında, program sistem trayde çalışır ve buradaki ikona sağ tıklayarak aşağıdaki menü görüntülenir.

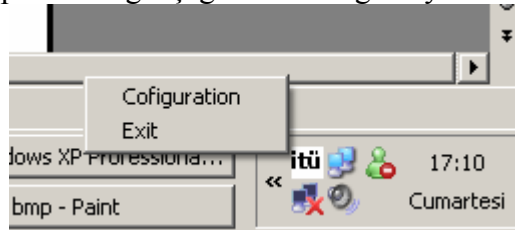


Windows İşletim Sisteminde Sistem Trayde Çalışan Program



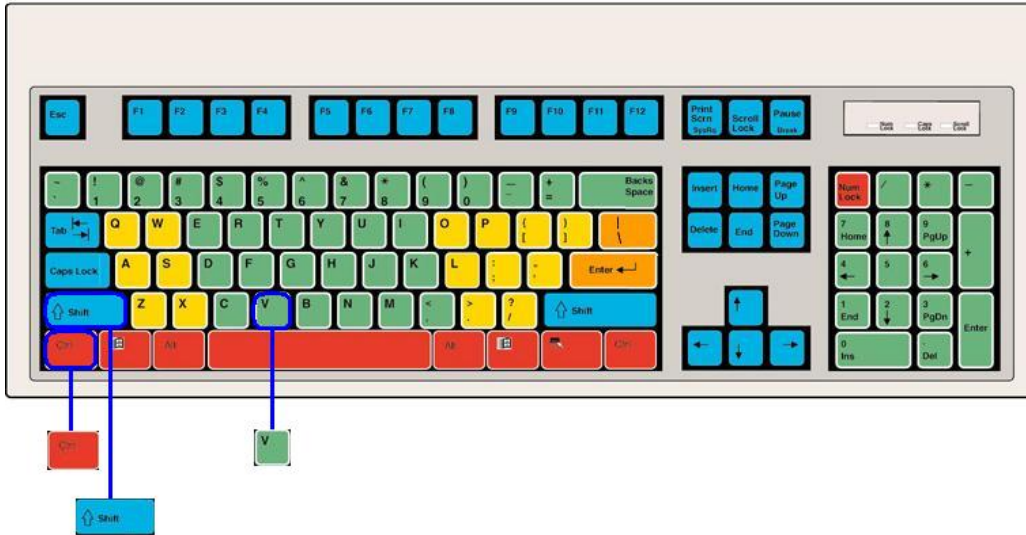
Linux İşletim Sisteminde Sistem Trayde Çalışan Program

Buradan ilgili ayarların yapılabileceği aşağıdaki konfigürasyon menülerine ulaşılır:



Windows İşletim Sisteminde Konfigürasyon Menüsü

Kullanıcının uzaktan kopyalama işlemini başlatabilmesi için kullanıcının Ctrl+Alt+C tuş kombinasyonunu kullanması gerekmektedir. Bu tuş kombinasyonuna aynı anda basıldığı durumda program tetiklenmekte, veri transferi gerçekleştirilmektedir.



Programı Tetikleyen Tuş Kombinasyonu(CTRL + SHIFT + V)

Kullanıcı Arayüz Katmanı Yapısı

Bu katmanın kullanıcıya sunmuş olduğu işlem yetenekleri, dolayısıyla kullanıcıdan programa aktarılacak veriler yukarıda belirtildiği şekildedir. Kullanıcıdan alınacak veriler doğrultusunda yapılması gereken işlem her iki işletim sistemi için yazılmış olan programda da aynıdır. Kullanıcı arayüzlerini ve kullanıcıdan gelen veriler doğrultusunda bir alt katmanla gerçekleşecek olan iletişimi sağlayan sınıf yapıları aşağıda açıklanmıştır.

Windows İşletim Sistemi İçin Kullanıcı Arayüz Katmanı

Sınıf Adı: Form1

Amacı: Kullanıcıdan kısa yollarla aldığı komutları uygulamak. Bu komutlar karşıdaki bilgisayardan metin veya dosya kopyalamak olabilir. Şu an için metin kopyalama fonksiyonu tam olarak çalışmakta ve de dosya kopyalama işlemin geliştirilmesi devam etmektedir. Forum yapısı ve de işleyiş şekli yeni eklenebilecek özellikleri uygundur.

Program kullanıcının başka ortam veya programlarda yapacağı kopyalama veya yapıştır işlemleri ile ortaklaşa çalışacağından arka planda çalışmaktadır.

Programın veri gönderim ve alım işlemlerini ayarlamak için görünür formlar kullanılmıştır.

Form sistem tray dediğimiz Windows ortamlarında sağ al köşede çalışmaktadır.(Bkz. Şekil 5.3)

Sınıf Değişkenleri:

```
private ContextMenu m_menu;
server se;
Client cli;
```

Form1 Sınıf Değişkenleri

m_menu: Kullanıcının ikona sağ tıkladığında çıkan menüyü ve de tıkladığında formun hangi metodu çağıracağını düzenler. Bunlar ayarlama ve de çıkış seçenekleridir. (Bkz. Şekil 5.4)

se: Diğer bilgisayardan gelen klipbord isteklerine cevap verir. İçyapısı daha sonra detaylı bir şekilde aktarılacaktır.

cli: Diğer bilgisayardan gelen klipbord isteğinde bulunur. İçyapısı daha sonra detaylı bir şekilde aktarılacaktır.

Sınıf Metotları:

Metodun Adı: Form1

Metodun Amacı:

Formun ön değerlerini vermek. Kısa yolları sağlayan gerekli, tuş kombinasyonlarını sistemin kayıt defterine kayıt etmek. ContextMenu'yu oluşturmak. Ve de Sever se nesnesini dinlemeye geçirmek.

Metodun Algoritması:

Sıra ile aşağıdaki komutları yürütür:

Siteme Tuş Kombinasyonlarının Kayıt Komutu:

```
bool success = HotKeyRegister.RegisterHotKey(Handle, 101,
HotKeyRegister.KeyModifiers.Control |
HotKeyRegister.KeyModifiers.Shift, Keys.T);
```

Tuş kombinasyonları kayıt işlemi

Eğer sistemde bu tuş kombinasyonu var ise geri dönüş değeri false olacaktır. Burada siteme bir kesme geldiğinde (geldiği kanal bellidir) 101 değerinden hangi tuş kombinasyonuna basıldığı anlaşılmakta.

ContextMenu ye seçenek isimlerini ve de tıklandığında hangi metotların çağırılacağını yükler.

```
m_menu.MenuItems.Add(0,new MenuItem("Cofiguration", new
System.EventHandler(Conf_Click)));
m_menu.MenuItems.Add(1,new MenuItem("Exit", new
System.EventHandler(Exit_Click)));
se.StartListen(11000);
```

Context Menu yüklenmesi

Ve server'dan dinlemeye başlar:

Metodun Adı: WndProc

Metodun Amacı: Kısayol tuşlarına basılıp basılmadığını kontrol etmek. Gelen kesmenin numarasına bakarak klipbord gönderimini sağlamak veya klipbord isteğinde bulunmak.

Metodun Algoritması:

İşletim sistemi bu metodu ilgili kanaldan bir mesaj geldiğinde uyandırır. (Kod gizli olduğundan çalışma mekanizması bilinmiyor.) Mesaj o an da koşan tüm programları ilgilendirme olasılığı olan bir mesajdır. Gelen mesaj bir tuş kombinasyonuna basıldığında oluşan bir mesaj olup olmadığına bakılır. Eğer tuş kombinasyonundan geliyorsa kendisine ait olup olmadığına bakılır. Kendisine aitse metin kopyama veya dosya kopyama ile ilişkili işlemleri gerçekleştirir. Gerçekleme işlemleri daha sonra detaylı bir şekilde anlatılacaktır. Sitemden gelen mesajın hangi tuşu kombinasyonuna ait olduğunu kayıt sırasında verdiğimiz numara değerinden anlıyoruz.

Metodun Adı: Form1_FormClosing

Metodun Amacı: Program kapatılırken işletim siteminden aldığımız tuş kombinasyonlarını isteme geri iade eder.

Metodun Algoritması: Kayıt defterinden kayıtları silerek gerçekler.

```
bool success = HotKeyRegister.UnregisterHotKey(Handle, 101);
success = HotKeyRegister.UnregisterHotKey(Handle, 102);
```

Kayıt defterinden kayıtların silinmesi

Burada eğer kayıt silinemez ise geri dönem değeri false olacaktır. Eğer false ise kullanıcıya bunu bildiren bir hata mesajı basar.

Metodun Adı: Exit_Click

Metodun Amacı: Sever ve Client'ın bağlantılarını kapatacak metodları çağırır ve de programı sonlandırır.

Metodun Algoritması:

Server se nin StopListen()

Client cli nin Disconnect()

Form Form1 in Close() metodlarını çağırır.

Metodun Adı: Conf_Click

Metodun Amacı: Ayarla işlemlerinin gerçekleştiği form2 yi yaratır ve de aktif duruma geçirir. Form2 daha sonra anlatılacaktır.

Sınıf Adı: Form2

Amacı: İlk olarak kişinin RF ile mi yoksa ağ üzerinden mi haberleşeceğini seçmesine olanak sağlar.

Bu forma ikonun üzerine sağ tıklayınca çıkan menüden Configuration seçeneğine tıklayarak ulaşabilirsiniz. Daha sonra karşınıza:

Haberleşme Yolu Seçimi

Burada RF veya internetten alıp yapacak şekilde programı ayarlayabilirsiniz. RF bölümü projenin RF Modülünün geliştirilmesi bitmediğinden şu aktif değildir. Ağ üzerinden iletişim gerçekleştirecek olan kullanıcının ayarlarını yapabildiği arayüz aşağıda verilmiştir.

Ağ üzerinden haberleşme ayarları

Burada Alım bölümüne klipbord isteğinde bulunulacak bilgisayarın IP si gönderim listesine ise klipbord isteklerine cevap verilecek bilgisayarların IP listesi bulunmaktadır. Kullanıcı bunları isteğine bağlı olarak değiştirebilir. Bu ayarlar IPstorer sınıfı vasıtası ile XML bir dokümanda tutulur.

Server ve de Client Sınıfları da XML ile iletişimi sağlayan IPstorer sınıfını kullanarak bu IP lerin kayıtlı olduğu dokümana ulaşır.

Metodun Adı: radioButton2_CheckedChanged

Amacı: Kullanıcı RF iletişim şeklinden internet iletişim şekline geçtiğinde gereken görünüm değişikliklerini gerçekler.

Metodun Algoritması:

Eğer internet seçeneği seçilirse ilgili panel görünür olur.

Gönderim listesi resetlenir.

IPStorer IPS nesnesi yaratılır.

XML dokümanından veriler çekilir.

İlk genel IP yani ID si 1 olan IP alım yapılacak bilgisayar IP si geri kalan IP ler ise gönderim listesini oluşturur.

Metodun Adı: button1_Click

Amacı: Gönderim listesine, eğer yazılı olan değer geçerli bir formatta yazmışsa IP ise, yazılı olan IP yi ekler ancak XML dokümanına ekleme işi başka bir butona aittir ve bu kullanıcıya iptal etme şansı tanır.

Metodun Adı: button2_Click

Amacı: Gönderim listesinde seçili olan değeri listeden kaldırır ama XML dokümanında güncellemeyi başka bir botun yapar.

Metodun Adı: button3_Click

Amacı: Kullanıcı gönderim listesinde ve de alım bölümünde değişiklikleri yaptıktan sonra bu butonu tıklar ve de XML dokümanına kaydetmiş olur.

Linux İşletim Sistemi İçin Kullanıcı Arayüz Katmanı

Sınıf Adı: ShowTray

Amacı: Programı KDE’de sağ alt köşede çalışır duruma getirmek, kullanıcının konfigürasyon dosyalarına erişmesi için ilgili menüleri yaratmak ve kullanıcıdan gelen konutlara göre konfigürasyon formunu açmak.

Sınıf Değişkenleri:

```
public static JMenuItem quit;
public static JMenuItem configuration;
```

ShowTray Sınıf Değişkenleri

quit: Kullanıcının ikona sağ tıkladığında önüne çıkacak menüde bulunan çıkış elemanı

configuration: Kullanıcının ikona sağ tıkladığında önüne çıkacak menüde bulunan konfigürasyon formunun görünür olmasını sağlayan elemanı

Sınıf Metotları:

Metodun Adı: ShowTray

Metodun Amacı:

Menünün oluşturulması ve bu menü elemanlarından herhangi bir tanesinin seçilmesi durumunda yapılacak işlemlerin belirlenmesi.

Metodun Algoritması:

Menüye teker teker menü elemanları ilave edilir. Ardından her eklenen eleman çağrıldığında çalıştırılacak olan fonksiyonlar belirlenir.

Sınıf Adı: Configuration

Amacı: Kullanıcıdan konfigürasyon bilgilerinin alınması, kullanıcıya kayıtlı bilgilerin gösterilmesi ve güncellenmesine izin verilmesi.

Sınıf Metotları:

Metodun Adı: Configuration

Metodun Amacı:

Konfigürasyon dosyasında halihazırda bulunan ayar bilgilerinin alınarak ekrana aktarılması.

Metodun Algoritması:

XMLParser nesnesinin readall methodunu çağırarak xml dokümanındaki verileri alır ve forma yerleştirir.

Ağ üzerinden haberleşme ayarları

Metodun Adı: jButton1ActionPerformed

Metodun Amacı:

Yeni bir ip adresinin dışa veri aktarımı için xml dosyasına eklenmesi

Metodun Algoritması:

XMLParser nesnesinin insertip methodunu çağırarak xml dokümanına yeni bir dışa aktarım IP'si ekler. Ardından JList1 nesnesine yeni bir eleman ekleyerek kullanıcının eklenen IP'yi görmesini sağlar.

Metodun Adı: jButton2ActionPerformed

Metodun Amacı:

Dosyadan bir dışa aktarım IP'si siler.

Metodun Algoritması:

Öncelikle JList1 nesnesinde kayıtlı ve seçili eleman olup olmadığını kontrol eder. Herhangi bir kayıtlı IP adresi varsa XML dosyasından silme aşamasına geçer. XMLParser nesnesinin removeip methodunu çağırarak xml dokümanından ilgili dışa aktarım IP'sini siler.

Metodun Adı: jButton3ActionPerformed

Metodun Amacı:

Yeni bir ip adresinin içe veri aktarımı için xml dosyasına eklenmesi

Metodun Algoritması:

Öncelikle JList2 nesnesinde kayıtlı eleman olup olmadığını kontrol eder. Herhangi bir kayıtlı IP adresi yoksa XML dosyasına ekleme aşamasına geçer. XMLParser nesnesinin insertip methodunu çağırarak xml dokümanına yeni bir içe aktarım IP'si ekler. Ardından JList2 nesnesine yeni bir eleman ekleyerek kullanıcının eklenen IP'yi görmesini sağlar.

Metodun Adı: jButton4ActionPerformed

Metodun Amacı:

Dosyadan bir içe aktarım IP'si siler.

Metodun Algoritması:

Öncelikle JList2 nesnesinde kayıtlı ve seçili eleman olup olmadığını kontrol eder. Herhangi bir kayıtlı IP adresi varsa XML dosyasından silme aşamasına geçer. XMLParser nesnesinin removeip methodunu çağırarak xml dokümanından ilgili içe aktarım IP'sini siler.

İletişim Katmanı Yapısı

Bu katman kullanıcı arayüz katmanından gelen verileri yorumlayarak ağ üzerinden ya da RF modül üzerinden haberleşme işlemini gerçekleştirir. Bunun yanında her çeşit konfigürasyon işleminin de bu katmanda yorumlanarak bir alt katmana gönderilerek gerçekleştirilmesi sağlanır.

Windows İşletim Sistemi İçin İletişim Katmanı

Sınıf Adı: IPDefinition

Amacı: Programda IP yi temsil eder. IP ve de ID alanları vardır.

Sınıf Adı: Client

Amacı: Karşı bilgisayara soket kullanarak bağlanma, klipbord isteğini iletme ve de klipbordtan gelen veriyi bilgisayarın kendi klipborduna kaydetme işlerini gerçekleştirir.

Sınıf Değişkenleri:

```
byte[] m_dataBuffer: genel verilerin tutulduğu tampon bölge.
IAsyncResult m_result: Asenkron iletişim için.(Metotlarda açıklanacak)
public AsyncCallback m_pfnCallBack: Asenkron iletişim için.
public Socket m_clientSocket: Bağlantı için soket.
string clipboard: Klipbord için kullanılan geçici alan.
```

Client sınıfı değişkenleri

Sınıf Metotları:

Metodun Adı: Connect

Amacı: XML dokümanından alım yapılacak bilgisayar soket bağlantısının yapılması ve de asenkron olarak karşı bilgisayardan gelecek veri için beklemeye geçilmesi.

Metodun Algoritması: Soket oluşturur:

```
m_clientSocket = new Socket(AddressFamily.InterNetwork,
SocketType.Stream, ProtocolType.Tcp);
```

Connect methodu algoritması

Bağlanacağı bilgisayarın IP sini XML dokümanından çeker bunun için en alt katmanda bulunan IPstore sınıfından yaratılan IPS nesnesini kullanır. Burada alım yapılacak IP nin Id si 1'dir. Bağlantıyı gerçekler:

```
m_clientSocket.Connect(ipEnd)
```

Connect methodu algoritması

Bağlantı gerçekleşmiş ise bağlantıdan gelecek verileri için dinlemeye başlar ancak bu asenkron bir dinleme olduğundan ilgili veri geldiğinde uyanma işlemi gerçekleşir.

```
if (m_clientSocket.Connected){ WaitForData(); }
```

Connect methodu algoritması

Eğer bilgisayara ulaşılamıyorsa kullanıcıya hata basar.

Metodun Adı: WaitForData

Amacı: Soketten asenkron dinlemek için gerekli atamaları yapar.

Metodun Algoritması: Bir AsyncCallback sınıfından nesne yaratır:

```
m_pfnCallBack = new AsyncCallback(OnDataReceived);
```

WaitForData methodu algoritması

Bu olay gerçekleştiğinde client nesnesinin metoddu olan OnDataReceived çağırılacaktır. Olayın tanımı ise aşağıdaki şekildedir:

```
m_clientSocket.BeginReceive(theSocPkt.dataBuffer, 0, theSocPkt.dataBuffer.Length, SocketFlags.None, m_pfnCallBack, theSocPkt);
```

WaitForData methodu algoritması

Bu şekilde soket asenkron dinlemeye geçer ve de bir data geldiğinde m_pfnCallBack nesnesi uyarır o da OnDataReceived metodunu çağırır.

Metodun Adı: OnDataReceived

Amacı: Gelen klipbord içerik verisini almak ve de bilgisayarın klipborduna aktarmak.

Metodun Algoritması:

Veriyi alır:

```
int charLen = d.GetChars(theSockId.dataBuffer, 0, iRx, chars, 0);
System.String szData = new System.String(chars);
```

OnDataReceived methodu algoritması

Daha sonra veriyi Klipborda taşır.

Ancak burada çok önemli bir sorun oluşur. .NET platformu optimizasyon yaptığından genellikle ve de özellik asenkron çalışan birimle MTA modunda çalışır. MTA modunda klipbord a ulaşmak güvenlik açısından sakıncalı olduğundan bu metot da doğrudan ulaşmaya izin vermez.

Bu güvenlik kısıdını aşmanın yolu ise bu metottan yaratılan bir STA ipliğidir.

```
clipboard = szData;
Thread thrd = new Thread(new ThreadStart(STAClipboard));
thrd.SetApartmentState(ApartmentState.STA);
thrd.Start();
thrd.Join();
```

OnDataReceived methodu algoritması

STAClipboard metotdu bizim için gidip STA modunda szData verisini clipborda kaydeder. Metotlar birbirlerini göremedikleri için ortak alan olarak string tipinden clipbord değişkenine atama yapılmıştır.

Metodun Adı: STAClipboard

Amacı: clipbord nesne değişkenin içindeki veriyi bilgisayarın klipborduna kaydetmek.

Metodun Algoritması:

Aşağıdaki komutu yürütür:

```
Clipboard.SetDataObject(clipboard, true);
```

STAClipboard methodu algoritması

Metodun Adı: SendMessage

Amacı: Parametre olarak gelen veriyi socket aracı ile kar bilgisayar iletir. Programda genel karşı bilgisayara clipbord isteği gönderirken kullanılır.

Metodun Algoritması:

Mesajı ASCII koduna çevirir ve mesajı yollar.

```
byte[] byData = System.Text.Encoding.ASCII.GetBytes(msg);
m_clientSocket.Send(byData);
```

SendMessage methodu algoritması

Metodun Adı: Disconnect

Amacı: Socket bağlantısını kapatmak.

Sınıf Adı: Server

Amacı: Bilgisayara socket kullanarak dinleme ve de istemci bilgisayara eğer ki gönderim listesinde ise Klipbord içeriği gönderme.

Sınıf Değişkenleri:

```
public AsyncCallback pfnWorkerCallBack: Asenkron iletişim için.
private Socket m_mainSocket: Bağlantıyı sağlayacak socket.
```

Server sınıfı değişkenleri

Sınıf Metotları:

Metodun Adı: StartListen

Amacı: Parametre olarak genel porttan asenkron dinlemeye başlar. Soketten veri geldiğinde asenkron olarak OnClientConnect metodunu çağıracak şekilde soketi ayarlar.

Metodun Algoritması: Soket yaratılır:

```
m_mainSocket = new Socket(AddressFamily.InterNetwork,
SocketType.Stream, ProtocolType.Tcp);
```

StartListen methodu algoritması

Sokete veri geldiğinde uyandırılacak metod aşağıdaki komutla atanır:

```
m mainSocket.BeginAccept(new AsyncCallback(OnClientConnect), null);
```

StartListen methodu algoritması

Metodun Adı: OnClientConnect

Amacı: Bağlanan istemcinin hangi istemci olduğunu saptar. Ve de veriyi almak için WaitForData(workerSocket, m_clientCount) metodunu çağırır. workerSocket burada soket, m_clientCount ise istemci numarasıdır.

Metodun Adı: WaitForData

Amacı: Soketten veri geldiğinde asenkron olarak hangi metodun çalışacağını belirlemek.

Metodun Algoritması: AsyncCallback nesnesi yaratır ve de OnDataReceived metoduna buna bağlar. (Bu bir olay tanımlar.)

```
pfnWorkerCallBack = new AsyncCallback(OnDataReceived)
```

WaitForData methodu algoritması

Veri geldiğinde bu AsyncCallback olayı gerçekleşecek şekilde soket ayarlanır:

```
soc.BeginReceive(theSocPkt.dataBuffer, 0, theSocPkt.dataBuffer.Length,
SocketFlags.None, pfnWorkerCallBack, theSocPkt);
```

StartListen methodu algoritması

Metodun Adı: OnDataReceived

Amacı: İstemciden istediği klipbord içeriğini gösteren veriyi almak. Eğer istemci bilgisayar gönderim listesinde ise klipbord içeriğini istemci bilgisayara göndermek.

Metodun Algoritması:

İsteği alır ve de gelen istek gönderim listesindeki bir makinadan gelip gelmediğini kontrol eder:

```
for (int i = 0; i < IDs.Length; i++){
    IPs[i] = IPS.Load(IDs[i]);
    if (IPs[i].ID != "1" &&
socketData.m_currentSocket.LocalEndPoint.ToString() == IPs[i].IP
+":11000")
        ok = true;
}
```

OnDataReceived methodu algoritması

Eğer listede varsa istemci bilgisayar klipbord içeriği gönderilir. Burada client sınıfında karşılaştığımız sorun tekrar ortaya çıkar. MTA modunda olan bu metottan bir STA modunda iplik yaratılır ve de klipborddaki değeri ortak kullanılan bir değişkene aktarılmasına kadar beklenir. Ve son olarak veri yollanır:

```
Thread thrd = new Thread(new ThreadStart(STAClipboard));
thrd.SetApartmentState(ApartmentState.STA);
thrd.Start();
thrd.Join();
byte[] msg = Encoding.ASCII.GetBytes(clipboard);
Socket workerSocket = (Socket)socketData.m_currentSocket;
workerSocket.Send(msg);
```

OnDataReceived methodu algoritması

Metodun Adı: STAClipboard

Amacı: Klipborda STA modunda ulaşarak veriyi ortak bir değişkene aktarmak ve böylelikle çağıran metot veriye ulaşabilir.

Ulaşım yöntemi:

```
clipboard =
Clipboard.GetDataObject().GetData(DataFormats.Text).ToString();
```

STAClipboard methodu algoritması

Linux İşletim Sistemi İçin İletişim Katmanı

Sınıf Adı: ConnectionListener

Amacı:

Tüm bağlantıların bir listede tutulmasını sağlar. Böylelikle bir server ipliğine gelmiş olan çok sayıda bağlantı aynı anda kolaylıkla işlenebilir.

Sınıf Değişkenleri:

```
private Vector<Connection> connections;
```

ConnectionListener sınıfı değişkenleri

Sınıf Metotları:

Metodun Adı: run

Amacı: Yeni bir iplik yaratmak ve yaratılan ipliği listeye eklemek. Gerekğinde listeden silmek.

Metodun Algoritması: Sürekli yoklama işlemiyle gelebilecek mesajları kontrol eder. Ayrıca bu esnada kapanmış olan bağlantıları da listeden çıkarır. Ayrıca işlemciyi sürekli kullanmamak için bekleme süreleri bulunur.

```
try { Thread.sleep(100); }
catch (Exception e) { e.printStackTrace(); }
```

ConnectionListener ipliği işlemciyi sürekli kullanmamak için önlemi

Sınıf Adı: Connection

Amacı:

Herhangi bir porttan kurulmuş olan bir bağlantıyı simgeler. Gelen ve giden paketleri denetler.

Sınıf Değişkenleri:

```
private Socket socket;
private Out out;
private In in;
```

Connection sınıfı değişkenleri

Sınıf Metotları:

Metodun Adı: run

Amacı: Sürekli olarak dinleme yapan bir iplik yaratarak bağlantıya gelebilecek paketlerin kontrolünü gerçekleştirir.

Metodun Algoritması: Herhangi bir bağlantı geldiğinde gelen bağlantının IP adresini XML dosyasından XML parser nesnesini kullanarak kontrol eder. Eğer XML dosyasında ilgili IP adresine izin verilmişse, TextTransfer nesnesinin getClipboardContents methodu ile klipbordan içeriğini karşı tarafa gönderir.

Sınıf Adı: Client

Amacı:

İstemciyi modelleyerek bağlantıların daha sağlıklı kurulmasının sağlanması

Sınıf Değişkenleri:

```
private Socket socket;
private Out out;
private In in;
```

Connection sınıfı değişkenleri

Sınıf Metotları:

Metodun Adı: Client

Amacı: Overload edilmiş bir kurucu fonksiyondur. Nesneye ait parametrelerin ilk değer atamalarını gerçekleştirir.

Metodun Adı: listen

Amacı: kurulmuş olan bağlantıdan gelebilecek verileri beklemek ve herhangi bir veri geldiğinde bu gelen veriyi klipborda atamak.

Metodun Adı: request

Amacı: Bağlantı kurulmuş olan serverdan klipbord içeriğini talep etmek.

Sınıf Adı: Server

Amacı:

Serverı modelleyerek bağlantıların daha sağlıklı kurulmasının sağlanması.

Sınıf Metotları:

Metodun Adı: Server

Amacı: Sürekli olarak bir bağlantı talebi bekler. Herhangi bir bağlantı talebi geldiğinde onun için yeni bir iplik yaratır ve yeni kurulan bağlantıyı connections vektörüne ekler.

Sınıf Adı: TextTransfer

Amacı:

Klipbordun kontrolünü sağlamak. İçeriğini okumak ya da içeriğini değiştirebilmek.

Sınıf Metotları:

Metodun Adı: setClipboardContents

Amacı: Parametre olarak gelen veriyi klipborda yerleştirmek

Metodun Adı: getClipboardContents

Amacı: Klipbordun içeriğini okuyarak geri döndürmek.

Kayıt Katmanı Yapısı

Windows İşletim Sistemi İçin Kayıt Katmanı

Sınıf Adı: IPStorer

Amacı: Kullanıcıdan alınan IP gönderim listesini ve de alım IP sini XML dokümanında saklamak. XML dokümanında bulunan IP'leri çağrılan birime iletmek, güncellemek, yeni IP'ler eklemek ve silmek.

Sınıf Değişkenleri:

```
private XmlDocument MyDoc: XML dokümanını yönetmek için.
private string UserFileName: XML dokümanın adını tutar.
```

STAClipboard methodu algoritması

Sınıf Metotları:

Metodun Adı: IPStorer

Amacı: MyDoc değişkenine IP'lerin tutulduğu bu dosya yüklenir. Eğer aranan dosya bulunamıyorsa yaratılıp uygun formata getirilir, var olan dosya bozuk ise uygun formata getirilir.

Metodun Adı: Add

Amacı: Parametre olarak gelen IP yi XML dokümanın içine kaydeder. Ve eşsiz bir ID numarası verir.

Metodun Algoritması: XML dokümanın son elemanı okunarak anahtar olarak kullanılan ID değerlerinden en son eklenmiş olanı (en büyük olanı) okur. XmlElement tipinden yaratılan yeni nesnenin ID özelliğine dosyadan okunanın bir fazlası ve IP eklenir.

Metodun Adı: DeleteIP

Amacı: XML dokümanı içerisinde bulunan bir IP kaydın silinmesi

Metodun Algoritması: XML dokümanından ID'si giriş parametresi olarak verilen değer olan kayıt XmlDocument nesnesinin metotları kullanılarak bulunur ve silinir.

Metodun Adı: Load

Amacı: XML dokümanı içerisinde bulunan tüm kayıtlarının ID'lerinin toplanması

Metodun Algoritması: XML dokümanından, tüm kayıtların bir diziye alınması (XmlDocument sınıfının methodları vasıtasıyla) sonrasında dizinin her bir elemanının ID bilgisi bir string dizisine aktarılarak, bu dizi döndürülür. Eğer herhangi bir veri bulunmuyorsa, ID'yi temsil eden veri alanı olan tek elemanlık bir string dizisi yaratılarak döndürülür.

Metodun Adı: Load

Amacı: ID'si bilinen bir IP'nin XML dokümanından bulunup okunması.

Metodun Algoritması: XmlDocument nesnesinin methodları kullanılarak ID'si verilen kayda ait veri çekilir.

Metodun Adı: Remove

Amacı: Bütün IP kayıtlarının silinmesi

Linux İşletim Sistemi İçin Kayıt Katmanı

Sınıf Adı: XMLParser

Amacı: Kullanıcıdan alınan IP gönderim listesini ve de alım IP sini XML dokümanında saklamak. XML dokümanında bulunan IP'leri çağrılan birime iletmek, güncellemek, yeni IP'ler eklemek ve silmek.

Sınıf Metotları:

Metodun Adı: XMLParser

Amacı: Konfigürasyon bilgilerinin tutulduğu dosya yüklenir. Dosya bulunamadıysa yaratılır. Hatalıysa onarılır.

Metodun Adı: readall

Amacı: XML dosyasındaki tüm verileri okuyarak bir XML düğümü şeklinde geri döndürmek

Metodun Adı: insertip

Amacı: XML dosyasına giriş parametresi olarak gelen IP'yi eklemek

Metodun Adı: removeip

Amacı: XML dosyasından giriş parametresi olarak gelen IP'yi silmek

2.SONUÇ

Arada fiziksel ağ bağlantısı kullanılarak farklı işletim sistemlerinde dahil veri paylaşımı istenilen güvenlik ve hız kriterleri doğrultusunda gerçekleştirilmiştir. Kablosuz haberleşme işlemi bu kadar kolay olmamakla birlikte aynı güvenlik ve veri hızının elde edilmesinin yanında taşınabilir olması sebebiyle kullanıcıya kullanım kolaylığı sunmaktadır. Freescale firmasından temin edilen RF haberleşme birimleri teorik olarak 250kb/s hızda ortama bağlı olarak 10-75 m uzaklıktan veri aktarımını mümkün kılmaktadır. Proje kapsamında uygun algoritmalar geliştirilerek güvenlik ve hata kontrolleri istenilen seviyelerde yapılabilir.

Bir bilgisayardan diğer bilgisayara veri kopyalamak için günümüzde kullanılan yöntemler pek kullanışlı sayılmaz. Ağda paylaşma açma, e-mail atma gibi internet gerektiren yöntemler ya da USB bellek gibi donanımlar vasıtasıyla kopyalama işlemi gerçekleştirilmektedir. Bu proje ile kullanıcılar ilgili tuş kombinasyonu ile bir bilgisayardan diğerine veri aktarımını kolaylıkla gerçekleştirmektedir.